Fast Collaborative Graph Exploration

Dariusz Dereniowski, Yann Disser, Adrian Kosowski, Dominik Pajak, Przemyslaw Uznanski

Fast Collaborative Graph Exploration†

Dariusz Dereniowski[1] and Yann Disser[2] and Adrian Kosowski[3] and Dominik Pająk[3] and Przemysław Uznański[3]

[1] *Gdańsk University of Technology, Poland*
[2] *TU Berlin, Germany*
[3] *Inria Bordeaux Sud-Ouest, France*

We study the following scenario of online graph exploration. A team of k agents is initially located at a distinguished vertex r of an undirected graph. In each time step, each agent can traverse an edge of the graph. All vertices have unique identifiers, and upon entering a vertex, an agent obtains a list of the identifiers of all its neighbors. We ask how many time steps are required to complete exploration, i.e., to make sure that every vertex has been visited by some agent.

We consider two communication models: one in which all agents have global knowledge of the state of the exploration, and one in which agents may only exchange information when simultaneously located at the same vertex. As our main result, we provide the first strategy which performs exploration of a graph with n vertices at a distance of at most D from r in time $O(D)$, using a team of agents of polynomial size $k = Dn^{1+\varepsilon} < n^{2+\varepsilon}$, for any $\varepsilon > 0$. Our strategy works in the local communication model, without knowledge of global parameters such as n or D.

For any constant $c > 1$, we show that in the global communication model, a team of $k = Dn^c$ agents can always complete exploration in $D(1 + \frac{1}{c-1} + o(1))$ time steps, whereas at least $D(1 + \frac{1}{c} - o(1))$ steps are sometimes required. In the local communication model, $D(1 + \frac{2}{c-1} + o(1))$ steps always suffice to complete exploration, and at least $D(1 + \frac{2}{c} - o(1))$ steps are sometimes required.

---

1 Introduction

We are given a graph $G = (V, E)$ rooted at some vertex r. The number of vertices of the graph is bounded by n. A team of agents is charged with the task of exploring graph G, following the rules outlined in the Abstract. We assume that all agents have unique identifiers, are equipped with unbounded resources for local computations, and move in synchronous rounds.The agents' strategy is said to *explore* the graph G in t time steps if for all $v \in V$ there exists time step $s \leq t$ and an agent $g \in \mathcal{A}$, such that g is located at v in step s. Proposing an exploration strategy is relatively simple for a single agent. Assuming the agent is able to distinguish which neighboring vertices it has previously visited, there is no better systematic traversal strategy than a simple depth-first search of the graph, which takes $2(n-1)$ moves in total for a graph with n vertices. The situation becomes more interesting if multiple agents want to collectively explore the graph starting from a common location. If arbitrarily many agents may be used, then we can generously send n^D agents through the graph, where D is the distance from the starting vertex to the most distant vertex of the graph. At each step, we spread out the agents located at each node (almost) evenly among all the neighbors of the current vertex, and thus explore the graph in D steps.

In this paper we study this trade-off between exploration time and team size. A trivial lower bound on the number of steps required for exploration with k agents is $\Omega(D + n/k)$: for example, in a tree, some agent has to reach the most distant node from r, and each edge of the tree has to be traversed by some agent. We look at the case of larger groups of agents, for which D is the dominant factor in this lower bound. This complements previous research on the topic for trees [2, 4] and grids [6], which usually focused on the case of small groups of agents (when n/k is dominant).

---

†This work was initiated while A. Kosowski was visiting Y. Disser at ETH Zurich. Supported by ANR project DISPLEXITY and by NCN under contract DEC-2011/02/A/ST6/00201. The authors are grateful to Shantanu Das for valuable discussions and comments on the manuscript. The full version of this paper is available online at: `http://hal.inria.fr/hal-00802308`.

Another important issue when considering collaborating agents concerns the model that is assumed for the communication between agents. We need to allow communication to a certain degree, as otherwise there is no benefit to using multiple agents for exploration [4]. We distinguish between two communication models. In exploration *with global communication* we assume that, at the end of each step s, all agents have complete knowledge of the explored subgraph. In exploration *with local communication* two agents can exchange information only if they occupy the same vertex. Thus, each agent g has its own view on which vertices were explored so far, constructed based only the knowledge that originates from the agent's own observations and from other agents that it has met.

Related work

Collaborative online graph exploration has been intensively studied for the special case of trees. In [4], a strategy is given which explores any tree with a team of k agents in $O(D+n/\log k)$ time steps, using a communication model with whiteboards at each vertex that can be used to exchange information. This corresponds to a competitive ratio of $O(k/\log k)$ with respect to the optimum exploration time of $\Theta(D+n/k)$ when the graph is known. In [5] authors show that the competitive ratio of the strategy presented in [4] is precisely $k/\log k$. Also in [1] they prove a lower bound $\Omega(k/\log k)$ on the competitive ratio for the class of greedy algorithms. Another DFS-based algorithm, given in [1], has an exploration time of $O(n/k+D^{k-1})$ time steps, which provides an improvement only for graphs of small diameter and small teams of agents, $k=O(\log_D n)$. For a special subclass of trees called sparse trees, [2] introduces online strategies with a competitive ratio of $O(D^{1-1/p})$, where p is the density of the tree as defined in that work. The best currently known lower bound is much lower: in [3], it is shown that any deterministic exploration strategy with $k<\sqrt{n}$ has a competitive ratio of $\Omega(\log k/\log\log k)$, even in the global communication model. Both for deterministic and randomized strategies, the competitive ratio is known to be at least $2-1/k$, when $k<\sqrt{n}$ [4]. These lower bounds do not hold for larger teams of agents.

2 Tree exploration

We start our considerations by designing exploration strategies for the special case when the explored graph is a tree T rooted at a vertex r.

For any exploration strategy, the set of all encountered vertices (i.e., all visited vertices and their neighbors) at the beginning of step $s=1,2,3,\ldots$ forms a connected subtree of T, rooted at r and denoted by $T^{(s)}$. In particular, $T^{(1)}$ is the vertex r together with its children, which have not yet been visited. For $v\in V(T)$ we write $T^{(s)}(v)$ to denote the subtree of $T^{(s)}$ rooted at v. We denote by $L(T^{(s)},v)$ the number of leaves of the tree $T^{(s)}(v)$. Note that $L(T^{(s)},v)\le L(T^{(s+1)},v)$ because a leaf in $T^{(s)}(v)$ is either a leaf of the tree $T^{(s+1)}$ or the root of a subtree containing at least one vertex.

We are ready to give the procedure TEG (*Tree Exploration with Global Communication*).

Procedure TEG (tree T with root r, integer x) **at time step** s:
  Place x new agents at r.
  for each $v\in V(T^{(s)})$ which is not a leaf **do**: { determine moves of the agents located at v }
    Let $\mathcal{A}_v^{(s)}$ be the set of agents currently located at v.
    Denote by $v_1,v_2,\ldots,v_d$ the set of children of v.
    Let $i^*:=\arg\max_i\{L(T^{(s)},v_i)\}$. { v_{i^*} is the child of v with the largest value of L }
    Partition $\mathcal{A}_v^{(s)}$ into disjoint sets $\mathcal{A}_{v_1},\mathcal{A}_{v_2},\ldots,\mathcal{A}_{v_d}$, such that:
    (i) $|\mathcal{A}_{v_i}|=\left\lfloor\frac{|\mathcal{A}_v^{(s)}|\cdot L(T^{(s)},v_i)}{L(T^{(s)},v)}\right\rfloor$, for $i\in\{1,2,\ldots,d\}\setminus\{i^*\}$,
    (ii) $|\mathcal{A}_{v_{i^*}}|=|\mathcal{A}_v^{(s)}|-\sum_{i\in\{1,2,\ldots,d\}\setminus\{i^*\}}|\mathcal{A}_{v_i}|$.

    for each $i\in\{1,2,\ldots,d\}$ **do for each** agent $g\in\mathcal{A}_{v_i}$ **do move**$^{(s)}$ g to vertex v_i.
  end for
end procedure TEG.

Theorem 2.1 *For any fixed $c>1$ and known n, the online tree exploration problem with global communication can be solved in at most $D\cdot\left(1+\frac{1}{c-1}+o(1)\right)$ time steps using a team of $k\ge Dn^c$ agents by executing procedure* TEG *with parameter $x=O(n^c)$.*

Now we propose a strategy TEL (*Tree Exploration with Local Communication*). In the implementation of the algorithm we assume that whenever two agents meet, they exchange all information they possess about the tree. Thus, after the meeting, the knowledge about the explored vertices and their neighborhoods, is a union of the knowledge of the two agents before the meeting. Since agents exchange information only if they occupy the same vertex, at any time s, the explored tree $T^{(s)}$ may only partially be known to each agent, with different agents possibly knowing different subtrees of $T^{(s)}$.

Procedure TEL (tree T with root r, integer x) **at time step** s:
Place x new agents at r in state "exploring".
for each $v \in V(T^{(s)})$ which is not a leaf **do**: { determine moves of the agents located at v }
if $v \neq r$ **then for each** agent g at v in state "notifying" **do move**$^{(s)}$ g to the parent of v.
if v contains at least two agents in state "exploring" **and** agents at v do not have information of any agent which visited v before step s **then**:
{ send two new notifying agents back to the root from newly explored vertex v }
Select two agents g^*, g^{**} at v in state "exploring".
Change state to "notifying" for agents g^* and g^{**}.
move$^{(s)}$ g^* to the parent of v. { g^{**} will move to the parent one step later }
end if
Let $\mathcal{A}_v^{(s)}$ be the set of all remaining agents in state "exploring" located at v.
Denote by $v_1, v_2, \ldots, v_d$ all children of v, and by δ the distance from r to v.
$s' := \left\lfloor \frac{\delta+s}{2} \right\rfloor$. { s' is a time in the past such that $T^{(s')}(v)$ is known to the agents at v }
Let $i^* := \arg\max_i \{L(T^{(s')}, v_i)\}$. { v_{i^*} is the child of v with the largest value of L }
Partition $\mathcal{A}_v^{(s)}$ into disjoint sets $\mathcal{A}_{v_1}, \mathcal{A}_{v_2}, \ldots, \mathcal{A}_{v_d}$, such that:
(i) $|\mathcal{A}_{v_i}| = \left\lfloor \frac{|\mathcal{A}_v^{(s)}| \cdot L(T^{(s')}, v_i)}{L(T^{(s')}, v)} \right\rfloor$, for $i \in \{1, 2, \ldots, d\} \setminus \{i^*\}$,
(ii) $|\mathcal{A}_{v_{i^*}}| = |\mathcal{A}_v^{(s)}| - \sum_{i \in \{1,2,\ldots,d\} \setminus \{i^*\}} |\mathcal{A}_{v_i}|$.
for each $i \in \{1, 2, \ldots, d\}$ **do if** $|\mathcal{A}_{v_i}| \geq 2$ **then for each** agent $g \in \mathcal{A}_{v_i}$ **do move**$^{(s)}$ g to v_i.
for each $i \in \{1, 2, \ldots, d\}$ **do if** $|\mathcal{A}_{v_i}| = 1$ **then** change state to "discarded" for agent in $\mathcal{A}_{v_i}$.
end for
for each $v \in V(T^{(s)})$ which is a leaf **do move**$^{(s)}$ all agents located at v to the parent of v.
end procedure TEL.

In procedure TEL, all agents are associated with a state flag which may be set either to the value "exploring" or "notifying". Agents in the "exploring" state act similarly as in global exploration, with the requirement that they always move to a vertex in groups of 2 or more agents. Every time a group of "exploring" agents visits a new vertex, it detaches two of its agents, changes their state to "notifying", and sends them back along the path leading back to the root. These agents notify every agent they encounter on their way about the discovery of the new vertices. Although information about the discovery may be delayed, in every step s, all agents at vertex v know the entire subtree $T^{(s')}(v)$ which was explored until some previous time step $s' \leq s$. The state flag also has a third state, "discarded", which is assigned to agents no longer used in the exploration process.

The correctness of the definition of the procedure relies on the following lemma, which guarantees that for a certain value s' the tree $T^{(s')}(v)$ is known to all agents at v.

Lemma 2.2 *Let T be a tree rooted at some vertex r and let v be a vertex with distance δ to r. After running procedure* TEL *until time step s, all agents which are located at vertex v at the start of time step s know the tree $T^{(s')}(v)$, for $s' = \left\lfloor \frac{\delta+s}{2} \right\rfloor$.*

Theorem 2.3 *For any fixed $c > 1$, the online tree exploration problem can be solved in the model with local communication and knowledge of n using a team of $k \geq Dn^c$ agents in at most $D\left(1 + \frac{2}{c-1} + o(1)\right)$ time steps by executing procedure* TEL *with parameter $x = O(n^c)$.*

3 General graph exploration

Now we want to propose algorithm for general graphs. We will use procedures TEG and TEL. For any graph G we define a tree T and we show that algorithm exploring a tree T also explores graph G. Then we use TEG and TEL to explore tree T. Given a graph $G = (V,E)$ with root vertex r, we call $P = (v_0, v_1, v_2, \ldots, v_m)$ with $r = v_0$, $v_i \in V$, and $\{v_i, v_{i+1}\} \in E$ a *walk* of length $\ell(P) = m$. Note that a walk may contain a vertex more than once. We introduce the notation $P[j]$ to denote v_j, i.e., the j-th vertex of P after the root, and $P[0, j]$ to denote the walk $(v_0, v_1, \ldots, v_j)$, for $j \le m$. The last vertex of path P is denoted by $end(P) = P[\ell(P)]$. The concatenation of a vertex u to path P, where $u \in N(end(P))$ is defined as the path $P' \equiv P+u$ of length $\ell(P)+1$ with $P'[0, \ell(P)] = P$ and $end(P') = u$.

We now define the tree T with vertex set $\mathcal{P}$, root $(r) \in \mathcal{P}$, such that vertex P' is a child of vertex P if and only if $P' = P+u$, for some $u \in N(end(P))$. We first show that agents can simulate the exploration of T while in fact moving around graph G. Intuitively, while an agent is following a path from the root to the leaves of T, its location in T corresponds to the walk taken by this agent in G.

Lemma 3.1 *A team of agents can simulate the virtual exploration of tree T starting from root (r), while physically moving around graph G starting from vertex r.*

In order to execute procedures TEG and TEL we define function $L(T^{(s)}, v)$ such that it does not exceed n for any s, v. Then we can use TEG and TEL on tree T and we obtain following results.

Theorem 3.2 *For any $c > 1$, the online graph exploration problem with knowledge of n can be solved using a team of $k \ge Dn^c$ agents:*

- *in at most $D \cdot \left(1 + \frac{1}{c-1} + o(1)\right)$ time steps in the global communication model.*
- *in at most $D \cdot \left(1 + \frac{2}{c-1} + o(1)\right)$ time steps in the local communication model.*

For the case when we do not assume knowledge of (an upper bound on) n, we provide a variant of the above theorem which also completes exploration in $O(D)$ steps.

Theorem 3.3 *For any $c > 1$, there exists an algorithm for the local communication model, which explores a rooted graph of unknown order n and unknown diameter D using a team of k agents, such that its exploration time is $O(D)$ if $k \ge Dn^c$.*

4 Lower bounds

The graphs that produce the lower bound are a special class of trees. The same class of trees appeared in the lower bound from [4] for the competitive ratio of tree exploration algorithms with small teams of agents.

Theorem 4.1 *For every increasing function f, such that $\log f(n) = o(\log n)$, and every constant $c > 0$, there exists a family of trees $\mathcal{T}_{n,D}$, each with n vertices and height $D = \Theta(f(n))$, such that*

(i) for every exploration strategy with global communication that uses Dn^c agents there exists a tree in $\mathcal{T}_{n,D}$ such that number of time steps required for its exploration is at least $D\left(1 + \frac{1}{c} - o(1)\right)$,

(ii) for every exploration strategy with local communication that uses Dn^c agents there exists a tree in $\mathcal{T}_{n,D}$ such that number of time steps required for exploration is at least $D\left(1 + \frac{2}{c} - o(1)\right)$.

References

[1] P. Brass, F. Cabrera-Mora, A. Gasparri, and Jizhong Xiao. Multirobot tree and graph exploration. *Robotics, IEEE Transactions on*, 27(4):707 –717, 2011.

[2] Miroslaw Dynia, Jaroslaw Kutylowski, Friedhelm Meyer auf der Heide, and Christian Schindelhauer. Smart robot teams exploring sparse trees. In *MFCS*, pages 327–338, 2006.

[3] Miroslaw Dynia, Jakub Lopuszanski, and Christian Schindelhauer. Why robots need maps. In *SIROCCO*, pages 41–50, 2007.

[4] Pierre Fraigniaud, Leszek Gąsieniec, Dariusz R. Kowalski, and Andrzej Pelc. Collective tree exploration. *Networks*, 48(3):166–177, 2006.

[5] Yuya Higashikawa, Naoki Katoh, Stefan Langerman, and Shin-ichi Tanigawa. Online graph exploration algorithms for cycles and trees by multiple searchers. *Journal of Combinatorial Optimization*, pages 1–16, 2012.

[6] Christian Ortolf and Christian Schindelhauer. Online multi-robot exploration of grid graphs with rectangular obstacles. In *SPAA*, pages 27–36, 2012.